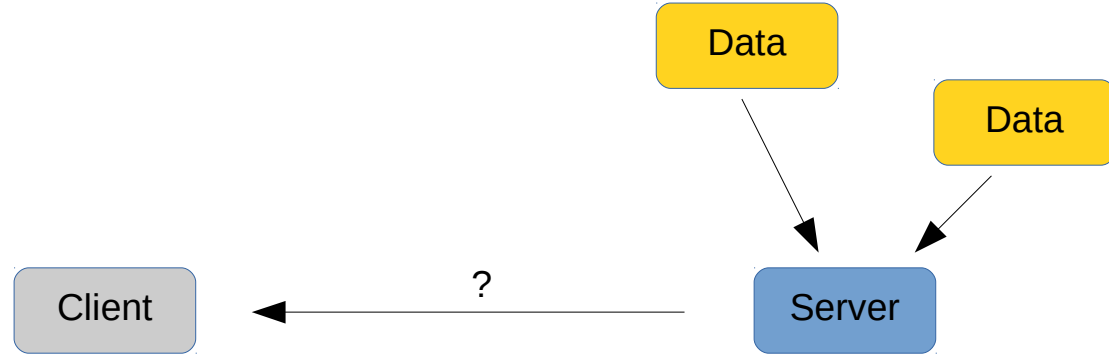
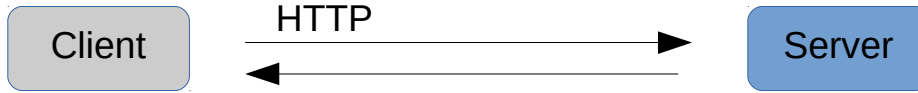


Putting MQTT in your toolkit

Sean Dague, Developer Advocate, IBM
June 9th 2018

Interaction Patterns

Client driven queries



Server driven data distribution

Real Time Event Strategies

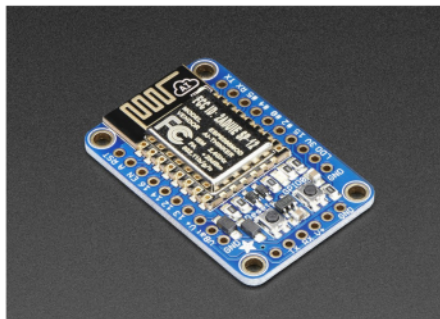
- Client Polling
 - Easy to implement
 - Really inefficient - 99+% noop
- HTTP Long Poll
 - Keep HTTP socket open, block on requests
 - Heavy server resource usage
- Websockets
 - Connected sockets in web browser
- Webhooks
 - Register callback URL
 - Need highly available "catcher" service
 - Only available for service owners
- MQTT / AMQP
 - Optimized pub/sub systems
 - Web client support requires bridging



What is MQTT?



- MQTT history
 - Created in 1999 by IBM & Cirrus Link, OASIS standard since 2013
 - Publish / Subscribe paradigm that requires a message broker
 - Designed for a "small code footprint" and "limited network bandwidth"
- Cloud IoT Services based on MQTT
 - Google IoT Core - <https://cloud.google.com/iot-core/>
 - IBM Watson IoT - <https://www.ibm.com/internet-of-things/>
 - Amazon IoT - <https://aws.amazon.com/iot-platform>
 - Microsoft Azure IoT - <https://azure.microsoft.com/en-us/services/iot-hub/>
- Open Source Servers / Clients everywhere
 - Notable - ESP8266
 - PubSubClient - Arduino MQTT implementation



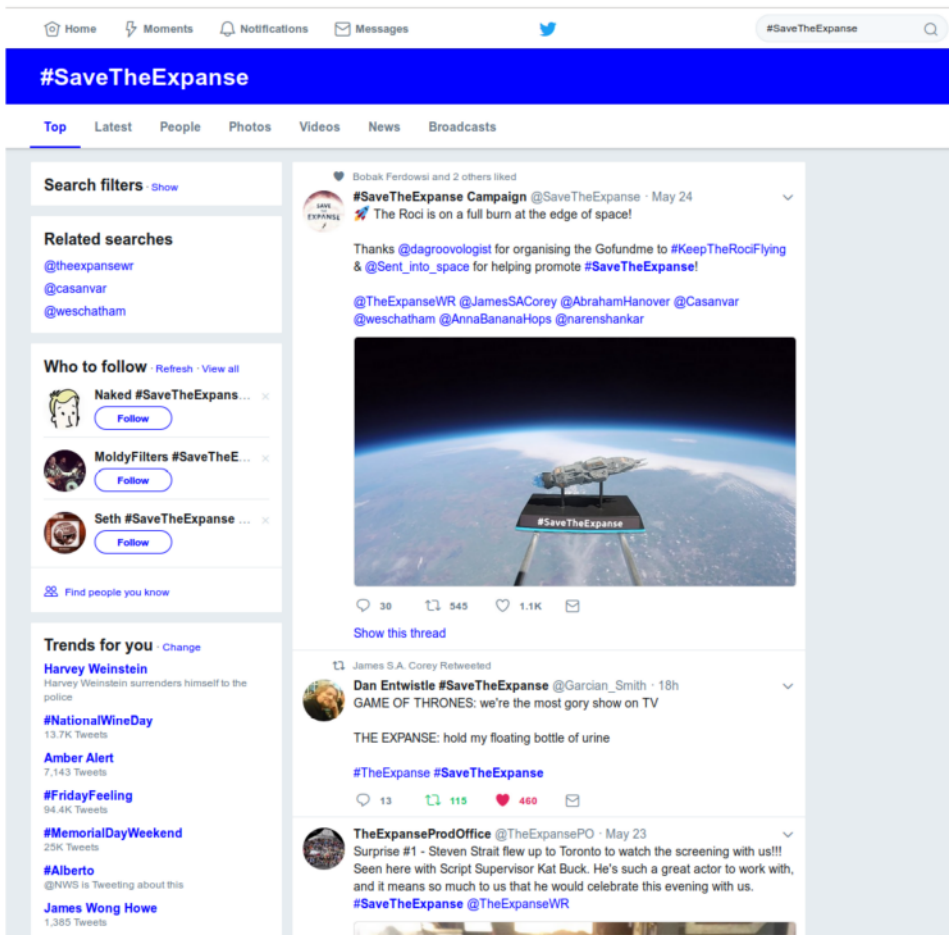
What is Pub / Sub

- Design Pattern
- Common message bus
- Everyone can publish to it
- Messages directed to topics
- Consumers subscribe to specific topics (possibly by wildcard)
- Great architecture for handling many to many interactions



What is Pub / Sub

- Design Pattern
- Common message bus
- Everyone can publish to it
- Messages directed to topics
- Consumers subscribe to specific topics (possibly by wildcard)
- Great architecture for handling many to many interactions



Mosquitto

- Open Source by Eclipse Foundation
- Written in C / highly performant
- Packaged for most Linux distros
- Includes CLI tools for pub/sub
- <https://mosquitto.org/>



MQTT Message Format

QoS	Retain	Topic	Payload
0,1,2	0,1	220 characters	Any content, up to 2 GB

- QoS - 0 best effort, 1 deliver at least once, 2 deliver exactly once
- Retain - content will be stored on the server, replayed on connect, defaults to not stored
- Topic - name for message, / are special
- Payload - anything, 2 GB payload limit

Note: no metadata on packets (like time sent), must put it in payload manually



Example sensible topic schema

{app name}/{location}/{device type}/{sensor type}/{reading type}

Allows the following subscriptions:

{app name}/{location}/# - see everything at a location

{app name}/+/{sensor type}/+ - see all particulate sensors

MQTT Wills

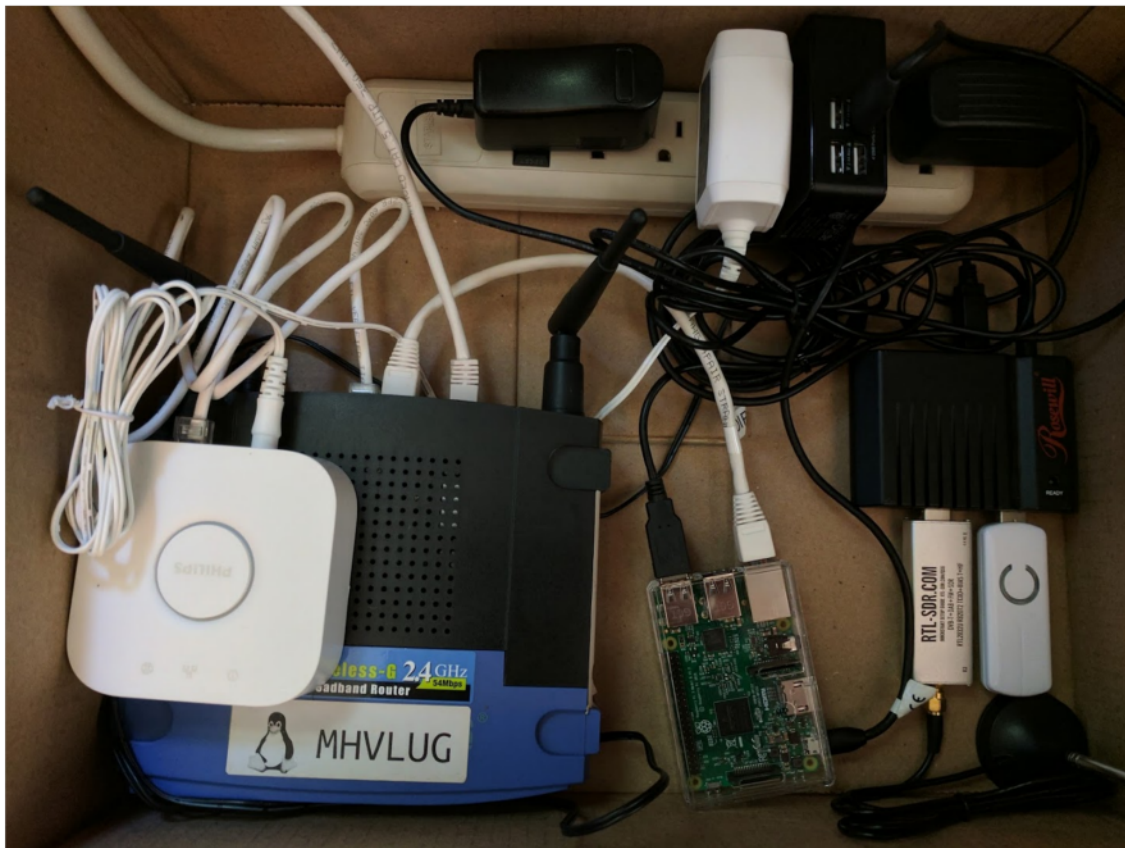
- Event based system - messages only sent when event happens
- If nothing is sent, is the client healthy with no new data, or did it disappear?
- Clients can set a "Will" on client connect
 - a message stored in the server that will be sent if the socket connection to the client breaks
 - building block for fault tolerance



Example Applications

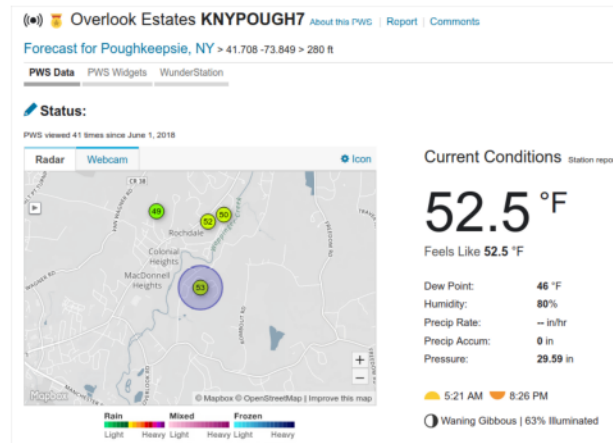


Home Weather Station



Bill of Materials

- Raspberry Pi 3
- RTL-SDR
- Oregon Scientific Sensors
- <http://github.com/sdague/arwn>

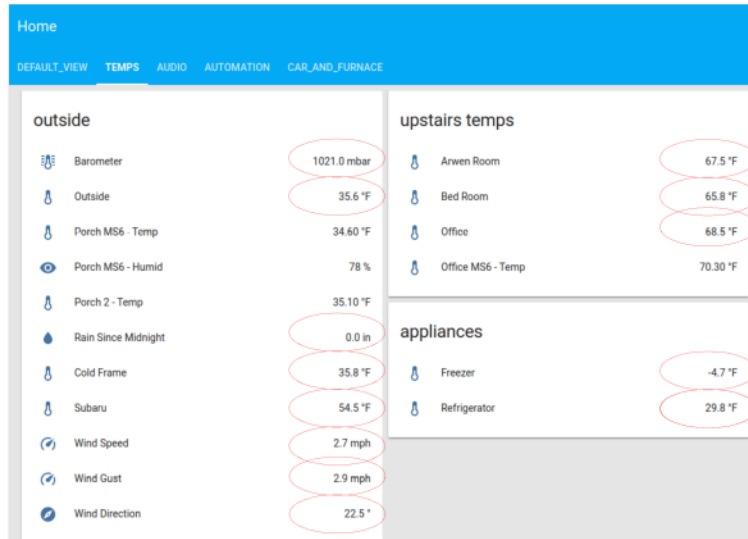




433 Mhz

0x0850022a9603814179

0x0d54012a96038141600304060079



mqtt:
broker: 10.42.0.3
sensor 1:
platform: arwn

```
arwn/temperature/Freezer {"bat": "LOW", "sensor_id": "6a:03", "humid": 53.0, "temp": -10.8, "dewpoint": -23.2, "units": "F", "timestamp": 1527695510}  
arwn/temperature/Arwen Room {"bat": "OK", "sensor_id": "ce:08", "humid": 54.0, "temp": 72.7, "dewpoint": 55.1, "units": "F", "timestamp": 1527695511}  
arwn/wind {"bat": "OK", "sensor_id": "33:00", "timestamp": 1527695512, "units": "mph", "gust": 0.9, "speed": 2.5, "direction": 315.0}
```

Home Assistant - Python 3 home automation

Home Assistant

Overview

Logbook

History

Log out

Developer tools

<>

Home

DEFAULT_VIEW

TEMPS

AUDIO

AUTOMATION

CAR_AND_FURNACE

outside

Barometer

1002.0 mbar

Outside

64.4 °F

Porch MS6 - Temp

55.70 °F

Porch MS6 - Humid

90 %

Porch 2 - Temp

57.50 °F

Rain Since Midnight

0.0 in

Subaru

54.1 °F

Wind Speed

0.0 mph

Wind Gust

0.0 mph

Wind Direction

270.0 °

upstairs temps

Anwen Room

68.4 °F

Bed Room

69.4 °F

Office

69.6 °F

Office MS6 - Temp

71.60 °F

appliances

Freezer

-6.3 °F

Refrigerator

36.7 °F

downstairs temps

Living Room - Temp

67.70 °F

Living Room - Humid

64 %

Family Room - Temp

65.70 °F

Hallway - Temp

67.70 °F

Basement

65.8 °F

Bomb Shelter

64.2 °F



Rain Gauge - Retain Topics

arwn/totals/rain	{"timestamp": 1528084804, "total": 70.78}
arwn/rain	{"timestamp": 1528084868, "total": 70.78}
arwn/rain/today	{"timestamp": 1528084868, "since_midnight": 0.00}
arwn/rain	{"timestamp": 1528091737, "total": 70.818}
arwn/rain/today	{"timestamp": 1528091737, "since_midnight": 0.04}
arwn/rain	{"timestamp": 1528092583, "total": 70.944}
arwn/rain/today	{"timestamp": 1528092583, "since_midnight": 0.16}
arwn/rain	{"timestamp": 1528107858, "total": 71.358}
arwn/rain/today	{"timestamp": 1528107858, "since_midnight": 0.58}

...

arwn/rain	{"timestamp": 1528171098, "total": 71.358}
arwn/rain/today	{"timestamp": 1528171098, "since_midnight": 0.58}

... rollover event

arwn/rain	{"timestamp": 1528171218, "total": 71.358}
arwn/totals/rain	{"timestamp": 1528171218, "total": 71.358}
arwn/rain/today	{"timestamp": 1528171218, "since_midnight": 0.0}



- Self emptying bucket
 - 3 increments added based on time between bucket dumps
- Reports Rain Total
 - (10ths of mm accumulator)
- Wunderground API
 - rainin - [rain inches over the past hour] -- the accumulated rainfall in the past 60 min
 - **dailyrainin** - [rain inches so far today in local time]

When Should I Charge My Car?



- Supports Time of Departure Charging
 - Be fully charged by a set time every day
- Time of Use metering at our home
 - Peak is 2 - 7pm Weekdays
 - Peak power costs 120%, off peak costs 89%
- What's the difference in power off peak?
 - What is generating the power?
 - What's the carbon intensity at different times?
- Can we make data available in real time?
 - Others might want to do things with this data

Market Data

Market & Operational Data

[Pricing Data](#)
[Power Grid Data](#)
[Load Data](#)
[Reports & Information](#)
[Postings by Date](#)
[Zone Maps](#)
[Charts & Graphs](#)
[Market Access Login](#)
[Custom Reports](#)
[TCC](#)
[ICAP](#)
[Distributed Energy Resource \(DER\)](#)
[Ancillary Services](#)
[Interregional Data](#)
[System Conditions](#)

Services

[Budget and Finance \[+\]](#)
[Customer Registration](#)
[Market Training \[+\]](#)
[Market Monitoring \[+\]](#)
[Reliability Compliance](#)

Support

[Customer Support](#)

Documents

[Tariffs](#)
[Manuals & Guides](#)
[Publications & Presentations](#)
[Technical Bulletins](#)
[Legal & Regulatory](#)
[Studies & Research](#)

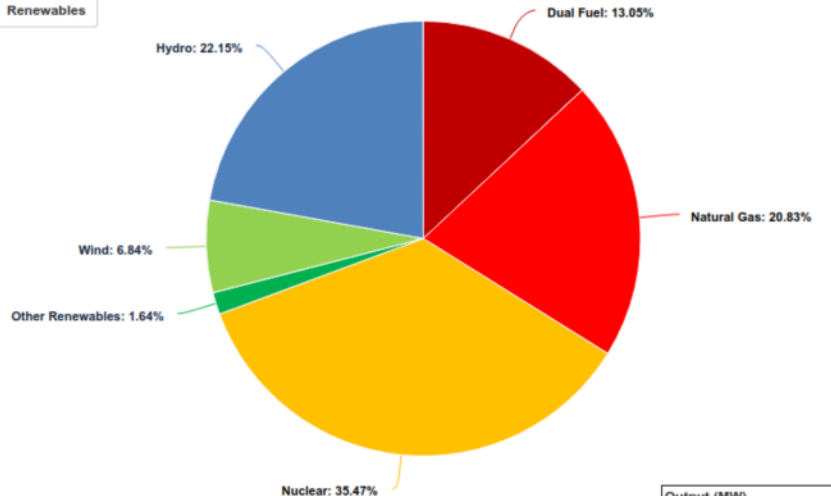
Data Graphs and Fuel Mix Chart

[Pricing Data](#)
[Power Grid Data](#)
[Load Data](#)
[Reports & Info](#)
[Postings by Date](#)
[Maps](#)
[Charts & Graphs](#)
[Market Access Login](#)
[Load](#)
[Load vs. LBMP](#)
[Flow](#)
[Daily Fuel](#)
[Current Fuel](#)
[Auto Refresh: On](#)
 Updated: 14:11

[All Fuels](#)
[Renewables](#)

Real-Time Fuel Mix as of 02/12/2018 2:00pm EST

Energy generated within New York State


[Current CSV](#)
[Historical CSV](#)

Output (MW)

Total 15,270

Renewables 4,679

Ancillary Service Prices

[Real-Time Market](#) P-68
[Day-Ahead Market](#) P-5

[NYISO Reference Bus LBMP](#) P-28
[NYISO Price Correction Logic](#) P-29

External RTO CTS Prices

[RTG vs. External RTO CTS Prices](#) P-42

Power Grid Data

Outages

[Real-Time Scheduled Outages](#) P-54A
[Real-Time Actual Outages](#) P-54B
[Day-Ahead Scheduled Outages](#) P-54C
[Outage Schedules](#) P-14
[Outage Schedules CSV](#) P-14B
[Generation Maintenance Report](#) P-15

Constraints

[Day-Ahead Limiting Constraints](#) P-511A
[Limiting Constraints](#) P-33

Interface Flows

[Internal & External Interface Limits & Flows](#) P-32
[Lake Erie Circulation - Day-Ahead](#) P-53B
[Lake Erie Circulation - Real-Time](#) P-34A

PARs

[PAR Schedules](#) P-53A
[PAR Flows](#) P-34

ATC/TTC

[ATC/TTC](#) P-8
[Long Term ATC/TTC](#) P-8A
[Transfer Limitations](#)

Load Data

Load Forecast/Commitment

[ISO Load Forecast](#) P-7
[Zonal Bid Load](#) P-59
[Weather Forecast](#) P-7A

Actual Load

[Real-Time Actual Load](#) P-58B
[Integrated Real-Time Actual Load](#) P-58C

Reports & Information

[Day-Ahead Market Daily Energy Report](#) P-30
[NYISO Capacity Report](#) P-60
[Balancing Market Advisory Summary](#) P-31
[Real-Time Events](#) P-35
[Real-Time Fuel Mix](#) P-63
[NYISO Operational Announcements](#) P-25
[Dispatcher Notices](#)

[NYISO Generation Economic Dispatch Analysis](#) P-61

[Generator Names](#) P-19
[Load Names](#) P-20
[Transmission Owner Information](#) P-16
[Subzones by Transmission Owner](#) P-23

[DOE EIA 930 - Same-Day Hourly Demand Posting](#) P-930A
[DOE EIA 930 - Daily Posting](#) P-930B

Real-Time Fuel Mix

CSV Files

[02-12-2018](#)
[02-11-2018](#)
[02-10-2018](#)
[02-09-2018](#)
[02-08-2018](#)
[02-07-2018](#)
[02-06-2018](#)
[02-05-2018](#)
[02-04-2018](#)
[02-03-2018](#)

Last Updated

02/12/18 14:10 EST
02/12/18 00:05 EST
02/11/18 00:05 EST
02/10/18 00:05 EST
02/09/18 00:05 EST
02/08/18 00:05 EST
02/07/18 00:05 EST
02/06/18 00:05 EST
02/05/18 00:05 EST
02/04/18 00:05 EST

Archived Files (zip format)

CSV Files

[02-2018](#)
[01-2018](#)
[12-2017](#)
[11-2017](#)
[10-2017](#)
[09-2017](#)
[08-2017](#)
[07-2017](#)
[06-2017](#)
[05-2017](#)
[04-2017](#)
[03-2017](#)
[02-2017](#)
[01-2017](#)
[12-2016](#)
[11-2016](#)
[10-2016](#)
[09-2016](#)
[08-2016](#)
[07-2016](#)
[06-2016](#)
[05-2016](#)
[04-2016](#)
[03-2016](#)
[02-2016](#)
[01-2016](#)
[12-2015](#)

Last Updated

02/12/18 14:10 EST
02/01/18 00:05 EST
01/01/18 00:05 EST
12/01/17 00:05 EST
11/01/17 00:05 EDT
10/01/17 00:05 EDT
09/01/17 00:05 EDT
08/01/17 00:05 EDT
07/01/17 00:05 EDT
06/01/17 00:05 EDT
05/01/17 00:05 EDT
04/01/17 00:05 EDT
03/15/17 11:23 EDT
03/15/17 11:05 EDT
03/15/17 11:24 EDT
03/15/17 10:36 EDT
11/01/16 00:05 EDT
10/01/16 00:05 EDT
09/01/16 00:05 EDT
08/01/16 00:05 EDT
07/01/16 00:05 EDT
06/01/16 00:05 EDT
05/01/16 00:05 EDT
04/01/16 00:05 EDT
03/01/16 00:05 EST
02/01/16 00:05 EST
01/01/16 00:05 EST

ny-power microservices architecture

NY ISO CSVs

Time Stamp	TZ	Fuel	Gen MW
05/09/2018 00:05:00	EDT	Dual Fuel	1400
05/09/2018 00:05:00	EDT	Natural Gas	2144
05/09/2018 00:05:00	EDT	Nuclear	4114

ny-power-pump

MQTT publish

```
ny-power/status/fuel-mix/updated {"ts": "05/09/2018 00:05:00"}
ny-power/upstream/fuel-mix/Nuclear {"units": "MW", "value": 4114, "ts": "05/09/2018 00:05:00"}
ny-power/upstream/fuel-mix/Dual Fuel {"units": "MW", "value": 1400, "ts": "05/09/2018 00:05:00"}
ny-power/upstream/fuel-mix/Natural Gas {"units": "MW", "value": 2144, "ts": "05/09/2018 00:05:00"}
```



```
ny-power/computed/co2 {"units": "g / kWh", "value": 135.088, "ts": "05/09/2018 00:05:00"}
```

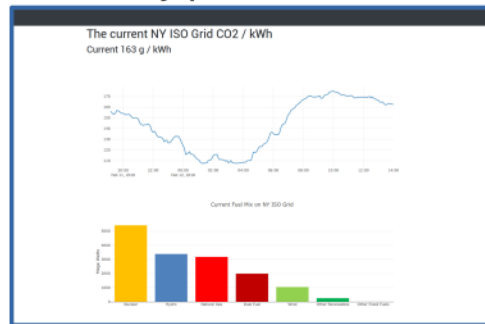
ny-power-mqtt (access with "mosquitto_sub -h 169.60.78.157 -t ny-power/# -v")

ny-power-influx

ny-power-archive

<http://ny-power.org>

ny-power-web



```
ny-power/archive/co2/24h {"units": "g / kWh", "values": [162.698, 163.928, 161.587 ... ], "ts": [...]}
```

ny-power topics

{app name}/{source}/{details}/{more details}

Allows the following subscriptions:

ny-power/upstream/fuel-mix/{fuel type}

ny-power/computed/co2

ny-power/archive/co2/24h




```

var client = new Paho.MQTT.Client("mqtt.ny-power.org", Number("80"), "client-" + Math.random());

// set callback handlers
client.onMessageArrived = onMessageArrived;

// connect the client
client.reconnect = true;
client.connect({onSuccess: onConnect});

// called when the client connects
function onConnect() {
    // Once a connection has been made, make a subscription and send a message.
    console.log("onConnect");
    client.subscribe("ny-power/computed/co2");
    client.subscribe("ny-power/archive/co2/24h");
    client.subscribe("ny-power/application/webui");
    client.subscribe("ny-power/upstream/fuel-mix/#");
}

```

```

// called when a message arrives
function onMessageArrived(message) {
    console.log("onMessageArrived:" + message.destinationName + message.payloadString);
    if (message.destinationName == "ny-power/computed/co2") {
        var data = JSON.parse(message.payloadString);
        $("#co2-per-kwh").html(Math.round(data.value));
        $("#co2-units").html(data.units);
        $("#co2-updated").html(data.ts);
    }
    if (message.destinationName.startsWith("ny-power/upstream/fuel-mix")) {
        fuel_mix_graph(message);
    }
    if (message.destinationName == "ny-power/archive/co2/24h") {
        var data = JSON.parse(message.payloadString);

```



The current NY ISO Grid CO2 / kWh

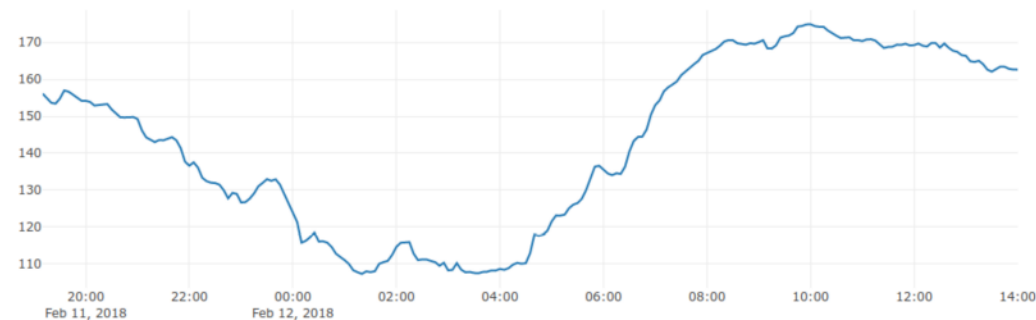
Current 163 g / kWh

<http://ny-power.org>

CO2 calculated from 2016 totals
(MW & Emissions Per fuel
source)

Provided as MQTT stream

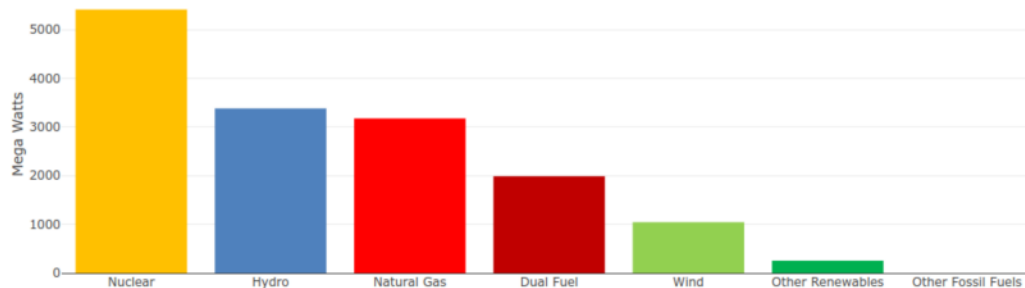
Answer: complete charging by
5am before load / NG starts
ramping up



<http://github.com/IBM/ny-power>

- Helm Kube application
- Core logic in python
- 5 pods

Current Fuel Mix on NY ISO Grid



@sdague

Put MQTT in your toolkit

- In a world awash in data, efficient event streams are critical
- Open event streams can be a public good
- MQTT has natural growth in the IoT space
- The Pub / Sub programming makes you think of problems in new ways

Thank You!

Twitter: **@sdague**

Email: sean.dague@ibm.com / sean@dague.net

Blog: <https://dague.net> - software engineering, open source projects, climate & energy

Get the code:

- arwn: <https://dague.net/arwn>
- ny-power project: <https://dague.net/ny-power>





CALL FOR CODE

COMMIT TO THE CAUSE. PUSH FOR CHANGE.

callforcode.org

Call for Code inspires developers to solve **pressing global problems** with **sustainable software solutions**, delivering on their vast potential to do good.

Bringing together NGOs, academic institutions, enterprises, and startup developers to compete build effective **disaster mitigation solutions**, with a focus on health and well-being.

The American Red Cross, and the **United Nations Human Rights Office** combine for the *Call for Code Award* to elevate the profile of developers.

Award winners will receive **long-term support** through the **Linux Foundation**, **financial prizes**, the **opportunity to present their solution to leading VCs**, and will deploy their solution through **IBM's Corporate Service Corps**. Developers will jump-start their project with dedicated **IBM Code Patterns**, combined with **optional enterprise technology** to build projects over the course of three months.

Judged by the world's most **renowned technologists**, the **grand prize** will be presented in **October** at an Award Event.